

## Usage

`eks-node-viewer` is a tool for visualizing dynamic node usage within a cluster. It was originally developed as an internal tool at AWS for demonstrating consolidation with Karpenter. It displays the scheduled pod resource requests vs the allocatable capacity on the node. It *does not* look at the actual pod resource usage.

```
5 nodes 125500m/127330m 98.6% cpu $5.324/hour $3886.520/month
515 pods (0 pending 515 running 515 bound)

ip-192-168-132-238.us-west-2.compute.internal cpu 100% (130 pods) c6a.8xlarge/$1.224 On-Demand ready
ip-192-168-93-114.us-west-2.compute.internal cpu 99% ( 34 pods) c6a.2xlarge/$0.306 On-Demand ready
ip-192-168-36-26.us-west-2.compute.internal cpu 96% (140 pods) c4.8xlarge/$1.591 On-Demand ready
ip-192-168-47-66.us-west-2.compute.internal cpu 99% (145 pods) c4.8xlarge/$1.591 On-Demand ready
ip-192-168-102-232.us-west-2.compute.internal cpu 100% ( 66 pods) c6a.4xlarge/$0.612 On-Demand ready

Press any key to quit
```

## Talks Using `eks-node-viewer`

- Containers from the Couch: Workload Consolidation with Karpenter
- AWS re:Invent 2022 - Kubernetes virtually anywhere, for everyone

## Installation

### Homebrew

```
1 brew tap aws/tap
2 brew install eks-node-viewer
```

**Manual** Please either fetch the latest release or install manually using:

```
1 go install github.com/awslabs/eks-node-viewer/cmd/eks-node-viewer@latest
```

Note: This will install it to your `GOBIN` directory, typically `~/go/bin` if it is unconfigured.

## Usage

```
1 Usage of ./eks-node-viewer:
2   -attribution
3       Show the Open Source Attribution
```

---

```
4  -context string
5      Name of the kubernetes context to use
6  -disable-pricing
7      Disable pricing lookups
8  -extra-labels string
9      A comma separated set of extra node labels to display
10 -kubeconfig string
11     Absolute path to the kubeconfig file (default "~/.kube/config")
12 -node-selector string
13     Node label selector used to filter nodes, if empty all nodes
14     are selected
15 -node-sort string
16     Sort order for the nodes, either 'creation' or a label name.
17     The sort order can be controlled by appending =asc or =dsc
18     to the value. (default "creation")
19 -resources string
20     List of comma separated resources to monitor (default "cpu")
21 -style string
22     Three color to use for styling 'good','ok' and 'bad' values.
23     These are also used in the gradients displayed from bad ->
24     good. (default "#04B575,#FFFF00,#FF0000")
25 -v      Display eks-node-viewer version
26 -version
27     Display eks-node-viewer version
```

## Examples

```
1  # Standard usage
2  eks-node-viewer
3  # Karpenter nodes only
4  eks-node-viewer --node-selector karpenter.sh/nodepool
5  # Display both CPU and Memory Usage
6  eks-node-viewer --resources cpu,memory
7  # Display extra labels, i.e. AZ
8  eks-node-viewer --extra-labels topology.kubernetes.io/zone
9  # Sort by CPU usage in descending order
10 eks-node-viewer --node-sort=eks-node-viewer/node-cpu-usage=dsc
11 # Specify a particular AWS profile and region
12 AWS_PROFILE=myprofile AWS_REGION=us-west-2
```

## Computed Labels

`eks-node-viewer` supports some custom label names that can be passed to the `--extra-labels` to display additional node information.

- `eks-node-viewer/node-age` - Age of the node

- 
- `eks-node-viewer/node-cpu-usage` - CPU usage (requests)
  - `eks-node-viewer/node-memory-usage` - Memory usage (requests)
  - `eks-node-viewer/node-pods-usage` - Pod usage (requests)
  - `eks-node-viewer/node-ephemeral-storage-usage` - Ephemeral Storage usage (requests)

## Default Options

You can supply default options to `eks-node-viewer` by creating a file named `.eks-node-viewer` in your home directory and specifying options there. The format is `option-name=value` where the option names are the command line flags:

```
1 # select only Karpenter managed nodes
2 node-selector=karpenter.sh/nodepool
3
4 # display both CPU and memory
5 resources=cpu,memory
6
7 # show the zone and nodepool name by default
8 extra-labels=topology.kubernetes.io/zone,karpenter.sh/nodepool
9
10 # sort so that the newest nodes are first
11 node-sort=creation=asc
```

## Troubleshooting

**NoCredentialProviders: no valid providers in chain. Deprecated.** This CLI relies on AWS credentials to access pricing data if you don't use the `--disable-pricing` option. You must have credentials configured via `~/aws/credentials`, `~/aws/config`, environment variables, or some other credential provider chain.

See credential provider documentation for more.

**I get an error of creating client, exec plugin: invalid apiVersion "client.authentication.k8s.io/v1alpha1"** Updating your AWS cli to the latest version and updating your kubeconfig should resolve this issue.

## Development

### Building

---

```
1 $ make build
```

Or local execution of GoReleaser build:

```
1 $ make goreleaser
```